This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-143 (cancelled)

Claim 144 (previously presented) A method of execution-instruction delegation between elemental processing resources of at least two different types, comprising:

obtaining an execution instruction, wherein the execution instruction is obtained at one of at least two of the elemental processing resources of a first type;

determining whether an operation-code within the execution instruction should be delegated to an other elemental processing resource of a second type different from the first type and shared by at least two elemental processing resources of the first type;

executing the execution instruction with the elemental processing resource of the first type, if the operation-code within the execution instruction should not be delegated to the other elemental processing resource; and

routing the execution instruction to the other elemental processing resource, if the operation-code within the execution instruction is for the other elemental processing resource.

Claim 145 (original) The method of claim 144, wherein the method is completed within a single processing cycle.

Claims 146-154 (cancelled)

Claim 155 (original) The method of claim 144, wherein the operation-code indicates a type of resource on which to execute.

Claim 156 (previously presented) The method of claim 144, wherein at least one elemental processing resource of the first type is an originating processing resource.

Claim 157 (previously presented) The method of claim 144, wherein the elemental processing resource of the first type is an integer processing unit.

Claim 158. (previously presented) The method of claim 144, wherein the elemental processing resource of the second type is a mathematical processing unit.

Claim 159 (cancelled)

Claim 160 (previously presented) The method of claim 144, wherein the elemental processing resource of the second type is a vector processing unit.

Claims 161-163 (cancelled)

Claim 164 (previously presented) The method of claim 144, wherein the elemental processing resource of the second type is an execution-instruction processing cache.

Claim 165 (original) The method of claim 144, further comprising routing the execution instruction through an execution-instruction signal router.

Claim 166 (cancelled)

Claim 167. (previously presented) The method of claim 144, wherein a first elemental processing resource executing a first individual thread may sleep while a second elemental processing resource executes delegated execution-instructions from the first individual thread.

Claim 168 (previously presented) The method of claim 144, wherein an execution-instruction signal causes various elemental processing resources dynamically to turn on and off to maintain a desired level of power draw while maximizing processing throughput.

Claim 169. (previously presented) The method of claim 144, further comprising generating an execution-instruction signal from elemental processing resources, wherein the execution-instruction signal from the elemental processing resources themselves shuts off processing resources while idling.

Claim 170 (previously presented) The method of claim 144, further comprising generating an execution-instruction signal from elemental processing resources, wherein an execution-instruction signal from processing resources themselves turn on processing resources when execution-instruction signal processing is required.

Claim 171. (previously presented) The method of claim 144, wherein the elemental processing resources are communicatively disposed on a same die.

Claim 172. (previously presented) The method of claim 171, wherein an execution-instruction signal router is on the same die with elemental processing resources.

Claims 173-1614 (cancelled)

Claim 1615 (previously presented) A method of execution-instruction delegation among multiple interdependent processing resources, the processing resources comprising multiple elemental processing resources of at least two IPU-type processing resources and at least one other type of elemental processing resource of either an MPU-type, an instruction delegating cache-type or a vector-type, that are configured to perform processing operations according instructions in an instruction set and are individually incapable of servicing at least one instruction in the instruction set, the method comprising:

receiving one of multiple execution-instructions from a thread at a first Instruction Processing Unit (IPU) of the at least two IPU-type processing resources;

processing the first execution-instruction using the first IPU;

receiving an other execution-instruction from the thread at the first IPU;

determining that the other execution-instruction from the thread can not be processed by the first IPU;

delegating the other execution-instruction from the thread to a processing resource, of a type other than an IPU-type processing resource, that is shared between the first IPU and a second IPU of the at least two IPU-type processing resources, the type of the processing resource other than the IPU-type processing resource being such that it and can process the other execution-instruction from the thread; and

maintaining the thread in the first IPU in a sleep state until an indicator that the processing of the other execution-instruction from the thread by the other type processing resource is returned.